Nonorthogonal Approximate Joint Diagonalization With Well-Conditioned Diagonalizers

Guoxu Zhou, Shengli Xie, *Senior Member, IEEE*, Zuyuan Yang, and Jun Zhang

***Abstract*—To make the results reasonable, existing joint diagonalization algorithms have imposed a variety of constraints on diagonalizers. Actually, those constraints can be imposed uniformly by minimizing the condition number of diagonalizers. Motivated by this, the approximate joint diagonalization problem is reviewed as a multiobjective optimization problem for the first time. Based on this, a new algorithm for nonorthogonal joint diagonalization is developed. The new algorithm yields diagonalizers which not only minimize the diagonalization error but also have as small condition numbers as possible. Meanwhile, degenerate solutions are avoided strictly. Besides, the new algorithm imposes few restrictions on the target set of matrices to be diagonalized, which makes it widely applicable. Primary results on convergence are presented and we also show that, for exactly jointly diagonalizable sets, no local minima exist and the solutions are unique under mild conditions. Extensive numerical simulations illustrate the performance of the algorithm and provide comparison with other leading diagonalization methods. The practical use of our algorithm is shown for blind source separation (BSS) problems, especially when ill-conditioned mixing matrices are involved.**

***Index Terms*—Approximate joint diagonalization, blind source separation (BSS), independent component analysis.**

I. Introduction

Approximate joint diagonalization is a problem of seeking a matrix $\mathbf{W}$ which makes $\mathbf{W}^H\mathbf{R}_k\mathbf{W}$ as diagonal as possible for all $\mathbf{R}_k \in \mathcal{R}$, where $\mathcal{R} = \{\mathbf{R}_k \in \mathbb{C}^{N\times N} | k = 1, 2, \ldots, K\}$ and $\mathbb{C}$ denotes the complex number set. Joint diagonalization has attracted much attention due to its wide-ranging applications in a variety of signal processing fields, such as blind source separation (BSS) [1]–[8], blind identification [9], [10], and blind wave beamforming [11]–[13]. Joint diagonalization is also the foundation of joint zero-diagonalization problems [7], [14].

As one of the most prominent applications of joint diagonalization, BSS is a problem of recovering the underlying sources only from their mixtures of sensors [1], [5], [11], [15]–[17]. Its linear instantaneous model is

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \qquad t = 1, 2, \ldots, T \tag{1}$$

where $\mathbf{s}(t)$ and $\mathbf{x}(t)$ denote the source vector and the observation vector at instant t, respectively. $\mathbf{A} \in \mathbb{C}^{N\times M}$ denotes the mixing matrix and $\mathbf{n}(t)$ models the additive noise (in this paper, $N \geq M$ is assumed). There are two steps for BSS methods based on joint diagonalization. First, a set of matrices relating to the mixing matrix $\mathbf{A}$ are generated. Typically, these matrices can be generated by using the second-order statistics (SOS). That is, $\mathbf{R}_k = \mathbf{R}_{\mathbf{x}}(\tau_k)$, where $\mathbf{R}_{\mathbf{x}}(\tau_k)$ is the correlation matrix of $\mathbf{x}(t)$ with time lag τ_k. Here, the time lags τ_k must be different for different k [1]. Second, jointly diagonalize the set using joint diagonalization algorithms. As a result, the output diagonalizer will be an estimate of $\mathbf{A}^{-1}$ up to scale factors and a permutation of columns (under some appropriate conditions, for example, see Proposition 4 in Section III). More details can be found in [1], [5], [11], [15], etc.

Joint diagonalization algorithms play a crucial role in the success of this type of separation methods. By far, a variety of algorithms have been proposed for the joint diagonalization problem [4]–[6], [8], [15], [18]–[22]. They are generally divided into two categories: orthogonal joint diagonalization (OJD) and nonorthogonal joint diagonalization (NJD). OJD algorithms restrict the diagonalizer to be orthogonal, and are applicable in BSS when the observations are prewhitened. However, because of some disadvantages in prewhitening phase in BSS [20], [22], [23], NJD has received increasing attention in recent years. Among the existing NJD algorithms, most are based on the following criterion of minimizing diagonalization error [5], [8], [15], [18], [21]:

$$\min F(\mathbf{W}) = \sum_{k\in\mathcal{I}_K} \operatorname{off}(\mathbf{W}^H\mathbf{R}_k\mathbf{W}) \tag{2}$$

where $\operatorname{off}(\mathbf{A}) = \sum_{i\neq j} \mathbf{A}_{ij}^2$ for any matrix $\mathbf{A}$ and $\mathbf{W} \in \mathbb{C}^{N\times M}$, $N \geq M$. For convenience $\mathbf{W}$ is called a diagonalizer hereafter. The index set $\mathcal{I}_K$ consists of the positive integers no greater than K. Although criterion (2) is straightforward, it cannot provide a desirable diagonalizer. Indeed, $\mathbf{W} = \mathbf{0}$ is a global minimum of (2) but it is not desirable. Consequently, NJD algorithms have two aspects: imposing proper constraints on (2) to make solutions reasonable, and designing an efficient optimization algorithm to solve the model. In this area, the fast Frobenius diagonalization (FFDIAG) algorithm updates $\mathbf{W}$ by a series of strictly diagonally dominant matrices, and thus $\mathbf{W}$ is invertible [5]. However, it is found that FFDIAG sometimes leads to unbalanced solutions, where certain rows and/or columns of the diagonalized matrices are implicitly weighted much less than others [21]. To avoid unbalanced solutions, the quadratic diagonalization (QDIAG) algorithm imposes norm constraints terms

Manuscript received November 21, 2008; revised June 24, 2009; accepted August 10, 2009. First published September 22, 2009; current version published November 04, 2009. This work was supported by the National Basic Research Program of China under 973 Program 2010CB731800, by the National Natural Science Foundation of China (NSFC) under Grants U0635001, U0835003, 60874061, and 60774094, and by the Distinguished Young Scholars of NSFC under Grant 60625102.

G. Zhou, S. Xie, and Z. Yang are with the South China University of Technology, Guangzhou 510641, China (e-mail: zhou.guoxu@mail.scut.edu.cn; adshlxie@scut.edu.cn; yangzuyuan@yahoo.com.cn).

J. Zhang is with the School of Electronics and Information Engineering, Beihang University, Beijing 510641, China (e-mail: buaazhangjun@vip.sina.com).

Digital Object Identifier 10.1109/TNN.2009.2030586

on diagonalizers

$$\left(\mathbf{W}^H\mathbf{C}^{(0)}\mathbf{W}\right)_{ii} = 1, \qquad i \in \mathcal{I}_M \tag{3}$$

where the matrix $\mathbf{C}^{(0)}$ is positive-definite symmetric and may be excluded from $\mathcal{R}$ [21]. However, Li *et al.* pointed out that QDIAG cannot avoid degenerate solutions [22], and then proposed a new criterion which is free of degenerate solutions

$$\min \sum_{k\in\mathcal{I}_K} \mathrm{off}(\mathbf{W}^H\mathbf{R}_k\mathbf{W}) - \beta \log|\det \mathbf{W}| \tag{4}$$

where $\mathbf{W} \in \mathbb{C}^{N\times N}$ and $\beta > 0$. Based on (4), the Fajd algorithm was developed. Fajd performs well when the set $\mathcal{R}$ is far away from being exactly jointly diagonalizable. However, if $\mathcal{R}$ is exactly jointly diagonalizable, i.e., there exists a matrix $\mathbf{W}$ such that $\sum_{k\in\mathcal{I}_K} \mathrm{off}(\mathbf{W}^H\mathbf{R}_k\mathbf{W}) = 0$, minima of (4) do not exist. Also, Fajd uses the inverse matrix of a matrix $\mathbf{Q}$ [defined later by (8)] which is possibly degenerate. All of these lead to reduced reliability of Fajd.

Subspace fitting formulation is also employed by some other algorithms, for instance, the alternating columns/diagonal centers (ACDC) algorithm [20]

$$\min E(\mathbf{V}, \mathbf{\Lambda}) = \sum_{k\in\mathcal{I}_K} \|\mathbf{R}_k - \mathbf{V}\mathbf{\Lambda}_k\mathbf{V}^H\|_F^2 \tag{5}$$

where $\mathbf{\Lambda}_k$ are diagonal matrices. Equation (5) requires some constraints for practical use. For example, Yeredor has considered the unit-norm constraints on the columns of $\mathbf{V}$ [20]. Degerine and Kane studied the relation between (2) and (5), and derived the QDIAG algorithm from another perspective (named LSB in their work) [15], [24]. For the LSB algorithm, the first matrix in $\mathcal{R}$ is required to be positive definite. If all the matrices in $\mathcal{R}$ are positive definite, another alternative has been proposed by Pham [19]. Generally speaking, these restrictions on $\mathcal{R}$ will limit the applications of NJD algorithms more or less.

In brief, all NJD algorithms attempt to provide *good* diagonalizers. But what is a *good* diagonalizer? In our point of view, a *good* diagonalizer should have as small condition numbers as possible when it minimizes the diagonalization error. Particularly, it must not be degenerate.

Significance of the condition number of a diagonalizer will be discussed in Section II, where it can be seen that the foregoing various constraints on $\mathbf{W}$ can be uniformly expressed in terms of condition number. Then, a new NJD algorithm, named FlexJD, is proposed in this paper. A major merit of FlexJD is that its output diagonalizers on the one hand achieve minimum diagonalization errors, and on the other hand, have flexible and possibly small condition numbers. Meanwhile, degeneration solutions are avoided strictly. Also, FlexJD imposes few restrictions on the matrices to be diagonalized.

The following notation conventions are used in this paper. Lowercase and uppercase bold letters denote column vectors and matrices, respectively. $\mathbf{A}_{ij}$ is the ijth entry of the matrix $\mathbf{A}$, and $\mathbf{a}_i$ is the ith column of $\mathbf{A}$. $\overline{\mathbf{A}_i}$ is the matrix obtained by removing the ith column of $\mathbf{A}$. Superscript H denotes the conjugate transpose. $\mathbf{I}$ and $\mathbf{0}$ denote the identity matrix and the zero matrix/vector of proper size.

For any matrix $\mathbf{A} \in \mathbb{C}^{N\times M}$, $\mathrm{col}(\mathbf{A}) = \{\mathbf{A}\mathbf{x} | \mathbf{x} \in \mathbb{C}^{M\times 1}\}$ denotes the linear space spanned by the columns of $\mathbf{A}$. $\dim(\cdot)$ denotes the dimension of a linear space. $\mathbf{A}^\perp$ is the orthogonal complement of $\mathrm{col}(\mathbf{A})$, and $\mathrm{Null}(\mathbf{A}^H)$ is an orthogonal basis of $\mathbf{A}^\perp$.

The column vector $\mathrm{diag}(\mathbf{A})$ is defined by the diagonal elements of the matrix $\mathbf{A}$ and $\mathrm{Diag}(\mathbf{a})$ is the diagonal matrix whose diagonal elements are given by the vector $\mathbf{a}$. $\mathrm{Ddiag}(\mathbf{A}) = \mathrm{Diag}(\mathrm{diag}(\mathbf{A}))$.

The determinant and condition numbers of a square matrix $\mathbf{A}$ are denoted by $\det \mathbf{A}$ and $\mathrm{cond}(\mathbf{A})$, respectively. □

The remainder of this paper is organized as follows. The issue of the condition number of a diagonalizer is raised in Section II. Section III is devoted to the development of the new algorithm, including convergence analysis and uniqueness of solutions. Simulations are presented in Section IV and finally conclusions are made in Section V.

II. Condition Number of a Diagonalizer

A. Why Condition Number Should be Considered

First, the diagonalizer $\mathbf{W}$ should be nondegenerate, i.e., $\mathrm{cond}(\mathbf{W}) < +\infty$. If not, model (2) becomes trivial since $\mathbf{W} = 0$ is a global minimum of it. However, free of degeneration is not enough, especially for BSS problems. As a matter of fact, $\mathbf{W}$ can have a very large condition number even if it is nondegenerate. A well-conditioned diagonalizer is significant to BSS: 1) it improves the numerical stability and 2) it makes the results more reasonable. In fact, the source vector $\mathbf{s}(t)$ is the solution to the linear system (1) and $\mathbf{W}$ is an estimate of $\mathbf{A}^{-1}$ (under some appropriate conditions). Therefore, if $\mathbf{W}$ is well conditioned, $\mathbf{s}(t)$ is not so sensitive to the noise $\mathbf{n}(t)$. Regarding 2), recall that two situations cause $\mathbf{W}$ to have a large condition number. The first is that one or more columns have extremely small norm compared with others (which coincides with the concept of unbalanced solutions reported in [21]); the second is that some columns are nearly linearly dependent. In BSS, the latter means that some outputs are nearly repeated signals and should be strictly avoided. We see that if $\mathbf{W}$ is well conditioned, trivial solutions, unbalance solutions, and degenerate solutions are totally avoided. Therefore, the condition number of a diagonalizer deserves attention.

We use two steps to control the condition number of $\mathbf{W}$.

B. Norm Constraints on Columns and the Condition Number

Norm constraints on the columns of $\mathbf{W}$ are simply indispensable, since they can not only eliminate ambiguities in NJD [15], but also affect the condition number of $\mathbf{W}$. However, it cannot ensure a well-conditioned diagonalizer [22]. Hence, norm constraints on columns are necessary but not sufficient for a well-conditioned diagonalizer.

Recall that QDIAG only imposes norm constraints on diagonalizers. One may question why QDIAG sometimes yields well-conditioned diagonalizers. The reason is as follows. For QDIAG, if $\mathbf{C}^{(0)} \in \mathcal{R}$, $\mathrm{off}(\mathbf{W}^H\mathbf{C}^{(0)}\mathbf{W}) \approx 0$ holds after (2) takes its minimum (assuming that this minimum sufficiently approaches zero). Together with (3), QDIAG actually imposes the constraint $\mathbf{W}^H\mathbf{C}^{(0)}\mathbf{W} \approx \mathbf{I}$ implicitly. In this case, QDIAG is

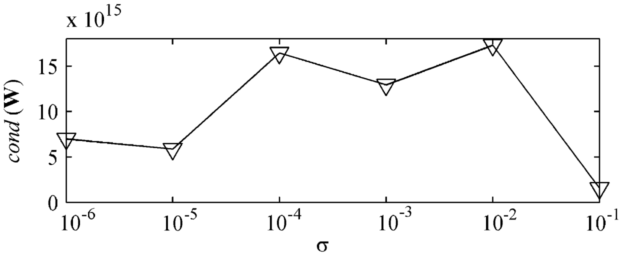
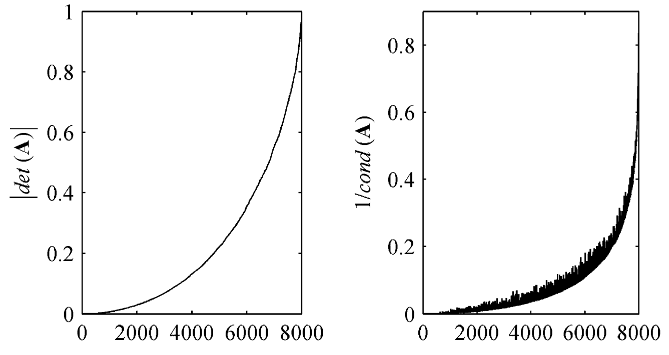


Fig. 1. Degeneracy of diagonalizers obtained by QDIAG if $\mathbf{C}^{(0)} \notin \mathcal{R}$. $\mathrm{cond}(\mathbf{W})$ for each σ has been averaged over 100 Monte Carlo trials.



(a) (b)

Fig. 2. Illustration of the relationship between the determinant and the condition number of a matrix under unit-norm constraints on columns. From the figure, the condition number of $\mathbf{A}$ decreases as $|\det(\mathbf{A})|$ increases proximately.

equivalent to the LSB algorithm, and they are essentially approximate OJD algorithms and consequently can provide well-conditioned diagonalizers. Unfortunately, if $\mathbf{C}^{(0)} \notin \mathcal{R}$, $\mathbf{C}^{(0)}$ is simply used to constraint the norm of the diagonalizers. Consequently, QDIAG often gives degenerate solutions; see Fig. 1. Hence, to QDIAG, $\mathbf{C}^{(0)}$ cannot be an arbitrary positive matrix if $\mathbf{C}^{(0)} \notin \mathcal{R}$. Generally, if none of the matrices in $\mathcal{R}$ are positive definite, $\mathbf{C}^{(0)}$ should be the closest positive-definite symmetric matrix to $\mathcal{R}$. (In Fig. 1, the matrices to be diagonalized are generated by (11) with $M = N = 5$, $K = 10$. The condition number of $\mathbf{W}$ for each σ has been averaged over 100 Monte Carlo trials, where $\mathbf{C}^{(0)} = \mathbf{I} \notin \mathcal{R}$ is selected and $\sigma = 10^{-6}, 10^{-5}, \ldots, 10^{-1}$.)

From now on, QDIAG exclusively means that $\mathbf{C}^{(0)} \in \mathcal{R}$.

C. Determinant and the Condition Number

First, $|\det\mathbf{W}|$ is not a good indicator for the condition number of a square matrix $\mathbf{W}$. To see this, fix some matrix $\mathbf{W}$ such that $0 < |\det\mathbf{W}| < +\infty$. $|\det(k\mathbf{W})| \to +\infty$ as $k \to +\infty$ but the condition number of $k\mathbf{W}$ is a constant. However, the case is quite different if the columns of $\mathbf{W}$ are normalized; see Fig. 2. We randomly generate 8000 matrices in $\mathbb{R}^{4\times 4}$ whose columns are normalized to be of unit norm. The values of $|\det\mathbf{W}|$ are plotted in Fig. 2(a) in ascending order and the corresponding values of $1/\mathrm{cond}(\mathbf{W})$ are plotted in Fig. 2(b). We see that a greater value of $|\det\mathbf{W}|$ is generally associated with a smaller condition number, although this relation is not strictly one-to-one.

The relation illustrated in Fig. 2 is not occasional. See the following inequality [25], [26]:

$$\mathrm{cond}(\mathbf{W}) < \frac{2}{|\det\mathbf{W}|}\left(\frac{\|\mathbf{W}\|_F}{\sqrt{N}}\right)^N = \frac{2}{|\det\mathbf{W}|}. \tag{6}$$

The equality is due to the unit-norm constraints. From (6), once $|\det\mathbf{W}|$ is maximized, the upper bound of $\mathrm{cond}(\mathbf{W})$ is just minimized. Consequently, it is possible to make $\mathbf{W}$ have a small condition number via maximizing the term $|\det\mathbf{W}|$. Furthermore, under the unit-norm constraints on the columns of $\mathbf{W}$, maxima of $|\det\mathbf{W}|$ exist.

Proposition 1: Given $\mathbf{W} \in \mathbb{C}^{N\times N}$, if $\mathbf{w}_i^H\mathbf{w}_i = 1$ for all $i \in \mathcal{I}_N$, then $|\det\mathbf{W}| \le 1$ holds. The equality holds if and only if $\mathbf{W}^H\mathbf{W} = \mathbf{I}$.

Proof: From the Hadamard inequality, $|\det\mathbf{W}|^2 = \det(\mathbf{W}^H\mathbf{W}) \le \det\mathrm{Ddiag}(\mathbf{W}^H\mathbf{W}) = 1$. The equality holds if and only if $\mathbf{W}^H\mathbf{W}$ is diagonal, i.e., $\mathbf{W}^H\mathbf{W} = \mathbf{I}$. □

As a result, under the unit-norm constraints on the columns, a well-conditioned $\mathbf{W}$ can be obtained by maximizing $|\det\mathbf{W}|$.

III. Presentation of the Proposed Algorithm

A. Model

Consider the following model:

$$\begin{aligned} \min F(\mathbf{W}) &= \sum_{k\in\mathcal{I}_K} \mathrm{off}(\mathbf{W}^H\mathbf{R}_k\mathbf{W}) \\ \max G(\mathbf{W}) &= \det(\mathbf{W}^H\mathbf{W}) \\ \text{s.t. } \mathbf{w}_i^H\mathbf{w}_i &= 1, \qquad i \in \mathcal{I}_M. \end{aligned} \tag{7}$$

From the analysis in Section II, model (7) is reasonable. Actually, model (4) is just the penalty function of (7) while the norm constraints are omitted. Model (7) has a hidden constraint of $G(\mathbf{W}) > 0$, since degenerate solutions should be avoided strictly.

Also note that we use $\det(\mathbf{W}^H\mathbf{W})$ instead of $|\det\mathbf{W}|$. We do this because the cases where $\mathbf{W}$ is square or tall (i.e., $N > M$) can be treated uniformly. Also, this will make the optimization easier.

B. Optimization Algorithm

For efficiency and simplicity reasons, alternate descent methods are adopted to optimize (7). Suppose $\overline{\mathbf{W}_i}$ is fixed and $\mathbf{w}_i$ is to be optimized. It is known that minimizing $F(\mathbf{W})$ is equivalent to minimizing the term $F_i(\mathbf{w}_i) = \mathbf{w}_i^H\mathbf{Q}_i\mathbf{w}_i$ [8], [15], [21], [22], where $\mathbf{Q}_i$ is defined by

$$\mathbf{Q}_i = \sum_{k\in\mathcal{I}_K}\left(\mathbf{R}_k\overline{\mathbf{W}_i}\,\overline{\mathbf{W}_i}^H\mathbf{R}_k^H + \mathbf{R}_k^H\overline{\mathbf{W}_i}\,\overline{\mathbf{W}_i}^H\mathbf{R}_k\right). \tag{8}$$

Remark 1: Note that the matrix $\overline{\mathbf{W}_i}\,\overline{\mathbf{W}_i}^H$ is rank-deficient and thus $\mathbf{Q}_i$ may be degenerate. In fact, $F_i(\mathbf{w}_i) = \mathbf{w}_i^H\mathbf{Q}_i\mathbf{w}_i$ is a part of the total diagonalization error. If the set of matrices is exactly jointly diagonalizable, $F_i(\mathbf{w}_i) = 0$ holds finally. Considering that $\mathbf{Q}_i$ is at least semipositive definite, $\mathbf{Q}_i$ is degenerate in this case.

As for $G(\mathbf{W})$, we have the following results.

Proposition 2: For given $\mathbf{W} \in \mathbb{C}^{N\times M}$, where $N \geq M$, if $\mathbf{w}_i^H\mathbf{w}_i = 1$, $i \in \mathcal{I}_M$, then:

1) $G(\mathbf{W}) = \det(\mathbf{W}^H\mathbf{W}) = \gamma\mathbf{w}_i^H\mathbf{B}_i\mathbf{w}_i$, where $\mathbf{B}_i = \mathbf{I} - \overline{\mathbf{W}}_i(\overline{\mathbf{W}}_i^H\overline{\mathbf{W}}_i)^{-1}\overline{\mathbf{W}}_i^H$ and $\gamma = \det(\overline{\mathbf{W}}_i^H\overline{\mathbf{W}}_i)$;

2) $\mathbf{w}_i^H\mathbf{B}_i\mathbf{w}_i = \mathbf{w}_i^H\mathbf{C}_i\mathbf{C}_i^H\mathbf{w}_i$, where $\mathbf{C}_i = \mathrm{Null}(\overline{\mathbf{W}}_i^H)$, which has the dimension of $N - M + 1$ and satisfies that $\mathbf{C}_i^H\mathbf{C}_i = \mathbf{I}$.

The proof is presented in Appendix I.

Note that γ is a constant irrelevant to $\mathbf{w}_i$. For convenience, we replace $G(\mathbf{W})$ by $\gamma G_i(\mathbf{w}_i)$. Therefore, $\mathbf{w}_i$ is updated by the solution of the following model:

$$\begin{aligned} \min F_i(\mathbf{w}_i) &= \mathbf{w}_i^H\mathbf{Q}_i\mathbf{w}_i \\ \max G_i(\mathbf{w}_i) &= \mathbf{w}_i^H\widetilde{\mathbf{C}_i}\mathbf{w}_i \\ \text{s.t. } \mathbf{w}_i^H\mathbf{w}_i &= 1 \end{aligned} \tag{9}$$

where $\mathbf{Q}_i$ is defined by (8) and $\widetilde{\mathbf{C}_i} = \mathbf{C}_i\mathbf{C}_i^H$.

Write $\mathbf{U} = [\mathbf{U}_0\ \mathbf{U}_1]$, where $\mathbf{U}_0$ and $\mathbf{U}_1$ are the eigenvector matrices of $\mathbf{Q}_i$ associated with the zero eigenvalues and the nonzero eigenvalues, respectively. $\mathbf{\Lambda}_1$ is the diagonal matrix whose diagonal elements are the nonzero eigenvalues. Model (9) is solved as follows.

Case 1) $\mathbf{Q}_i$ is invertible. In this case, a solution of (9) is given by $\max(G_i(\mathbf{w}_i)/F_i(\mathbf{w}_i))$. Thus, $\mathbf{w}_i$ is the unit-norm generalized eigenvector having the largest eigenvalue of $(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$.

Case 2) $\mathrm{col}(\mathbf{C}_i) \subseteq \mathrm{col}(\mathbf{U}_1)$ (or equivalently, $\mathbf{U}_0^H\mathbf{C}_i = \mathbf{0}$). Let $\mathbf{w}_i = \mathbf{U}_0\mathbf{x} + \mathbf{U}_1\mathbf{y}$. Thus, $G_i(\mathbf{w}_i) = \mathbf{y}^H\mathbf{U}_1^H\mathbf{C}_i\mathbf{C}_i^H\mathbf{U}_1\mathbf{y}$ and $F_i(\mathbf{w}_i) = \mathbf{y}^H\mathbf{\Lambda}_1\mathbf{y}$ for any $\mathbf{x}$. Thus, let $\mathbf{x} = \mathbf{0}$ and $\mathbf{y}$ be solved following the routine of Case 1.

Case 3) $\mathrm{col}(\mathbf{C}_i) \not\subseteq \mathrm{col}(\mathbf{U}_1)$. Let $\mathbf{w}_i = \mathbf{U}_0\mathbf{x} + \mathbf{U}_1\mathbf{y}$ again. It can be easily verified that $\lim_{\mathbf{y}\to\mathbf{0}}(G_i(\mathbf{w}_i)/F_i(\mathbf{w}_i)) = +\infty$ for $\mathbf{x} = \mathbf{x}^*$, where $\mathbf{x}^*$ is the solution to $\max g(\mathbf{x}) = \mathbf{x}^H\mathbf{U}_0^H\mathbf{C}_i\mathbf{C}_i^H\mathbf{U}_0\mathbf{x}$. Hence, let $\mathbf{y} = \mathbf{0}$ and $\mathbf{x} = \mathbf{x}^*$. $F_i(\mathbf{w}_i)$ reaches its global minimum 0 and $G_i(\mathbf{w}_i) = g(\mathbf{x}^*)$ is then maximized (meanwhile, $G_i(\mathbf{w}_i) > 0$), i.e., a nondegenerate diagonalizer with zero diagonalization error is obtained in this case. Finally, $\mathbf{x}^*$ is the unit-norm eigenvector having the largest eigenvalue of $\mathbf{U}_0^H\mathbf{C}_i\mathbf{C}_i^H\mathbf{U}_0$.[1] □

Remark 2: Among the above three cases, frequency of each case simply depends on the characteristic of the set of matrices to be diagonalized. In detail, if the set is exactly jointly diagonalizable, Cases 2 and 3 occur frequently after the algorithm almost converges (because diagonalization error gradually approaches zero); otherwise, if the set is far away from being exactly jointly diagonalizable, Case 1 dominates the algorithm.

Model (9) is a multiobjective quadratic optimization and the above procedure is denoted by $\mathbf{w}_i = \mathrm{MOQO}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$ for simplicity. It is worthwhile to note that the model $\mathrm{MOQO}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$ solution is not exactly equivalent to model (9). In fact, as a multiobjective optimization, model (9) generally has many incomparable solutions. These incomparable solutions are called Pareto optimal solutions [27], [28], which cannot be improved without hurting at least one criterion. $\mathrm{MOQO}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$ just attempts to give a Pareto optimal solution to model (9).

From the procedure of $\mathrm{MOQO}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$, $\det(\mathbf{W}^H\mathbf{W}) > 0$ strictly holds, and thus, degenerate solutions are avoided. The pseudocode of the proposed algorithm is displayed as follows.

```
While not terminated
    For $i = 1 : M$
        Update $\mathbf{Q}_i$ by (8), $\mathbf{C}_i = \mathrm{Null}(\overline{\mathbf{W}}_i^H)$, $\widetilde{\mathbf{C}_i} = \mathbf{C}_i\mathbf{C}_i^H$.
        $\mathbf{w}_i = \mathrm{MOQO}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$;
    End
End
```

The algorithm is terminated when $\mathbf{W}$ keep being unchanged. We name the new algorithm FlexJD, because its output diagonalizer is a tradeoff between the diagonalization error and the condition number, and this tradeoff is adaptive and flexible.

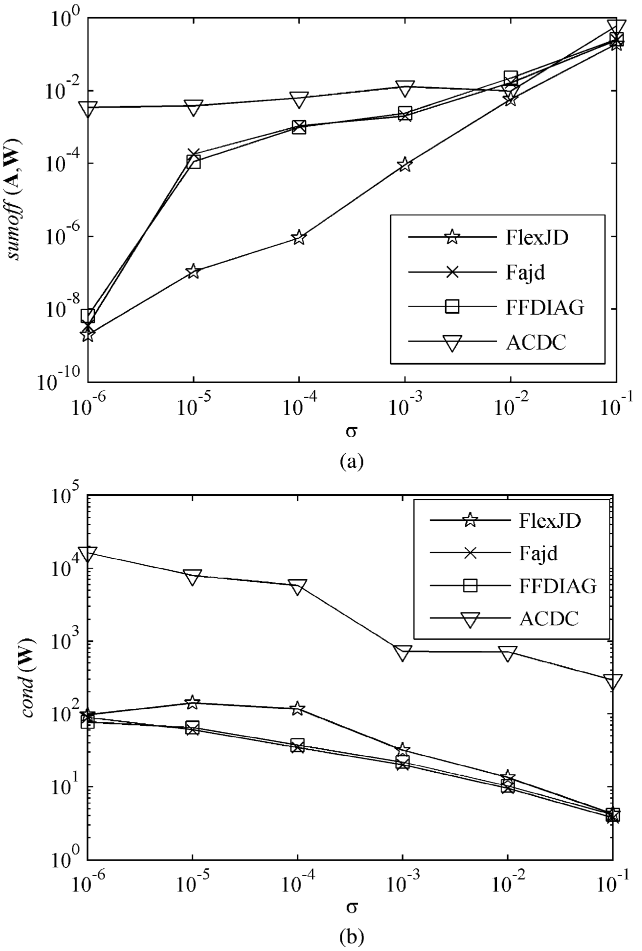
Remark 3: Note that $\dim(\overline{\mathbf{W}}_i) + \dim(\overline{\mathbf{W}}_i^{\perp}) = N$. Then, in the case of $M = N$, we have $\dim(\overline{\mathbf{W}}_i^{\perp}) = 1$, i.e., $\mathbf{C}_i$ is reduced to be a vector $\mathbf{c}_i$. If $\mathbf{Q}_i$ is invertible, it can be verified, from Case 1 in $\mathrm{MOQO}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$, that $\mathbf{w}_i = k\mathbf{Q}_i^{-1}\mathbf{c}$, which is just the normalized solution given by the Fajd algorithm. In this case, FlexJD and Fajd give the equivalent solution.

Remark 4: Tikhonov regularization methods can be used to solve (9) [29]: $\mathbf{w}_i$ is given by $\max H_i(\mathbf{w}_i) = \mathbf{w}_i^H\widetilde{\mathbf{C}_i}\mathbf{w}_i/\mathbf{w}_i^H(\mathbf{Q} + \varepsilon\mathbf{I})\mathbf{w}_i$, where $\varepsilon > 0$. (In our experiments, $\varepsilon = 10^{-6}$ is commonly recommended.) Note that $\mathbf{Q} + \varepsilon\mathbf{I}$ is positive definite, and thus, $\mathbf{w}_i$ is the unit-norm generalized eigenvector having the largest eigenvalue of $\widetilde{\mathbf{C}_i}$ and $\mathbf{Q} + \varepsilon\mathbf{I}$. We name this routine $\mathrm{Tikh}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i, \varepsilon)$. Simulations show that this routine almost gives the same result as MOQO.

C. Convergence and Uniqueness

Like many joint diagonalization algorithms, the FlexJD algorithm uses the so-called block coordinate descent methods in optimization [8], [15], [20]–[22], [30]. That is, a block of variables are optimized in each time whereas the others are fixed. Convergence analysis of this type of algorithms is often difficult and has been missed by most joint diagonalization algorithms [8], [20]–[22]. Fortunately, if the solutions of each subproblem are unique, the convergence respect to the entire parameter space is guaranteed [30]. From this fact, a primary result on the convergence of FlexJD can be presented. Suppose that $M = N$, then $\mathbf{C}_i$ is reduced to be a vector $\mathbf{c}_i$. In this case, if $\mathrm{Tikh}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i, \varepsilon)$ is implemented, it can be verified that the solution is unique and thus the algorithm is convergent. Because the solution of $\mathrm{Tikh}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i, \varepsilon)$ approaches to the solution of $\mathrm{MOQO}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$ as $\varepsilon \to 0$, we believe FlexJD based on $\mathrm{MOQO}(\widetilde{\mathbf{C}_i}, \mathbf{Q}_i)$ is also convergent. The following analysis is based on the assumption of $M = N$. Also, for simplicity, we assume that $\mathbf{R}_k = \mathbf{R}_k^H$ for $k \in \mathcal{I}_K$ and all the variables are real. (Extensions to complex-valued and asymmetric cases can be obtained in a similar manner.)

[1] Note that $G_i(\mathbf{w}_i)/F_i(\mathbf{w}_i) < +\infty$ for any $\mathbf{y} \neq \mathbf{0}$. Thus, Case 3 can be interpreted as the one that also solves the model $\max G_i(\mathbf{w}_i)/F_i(\mathbf{w}_i)$ and gives a solution such that $G_i(\mathbf{w}_i)/F_i(\mathbf{w}_i) = +\infty$ by setting $\mathbf{y} = \mathbf{0}$.



(a)

(b)

Fig. 3. (a) Evolution of diagonalization errors (in logarithmic scales) versus the level of noise. (b) Evolution of the condition numbers of diagonalizers (in logarithmic scales) versus the level of noise.

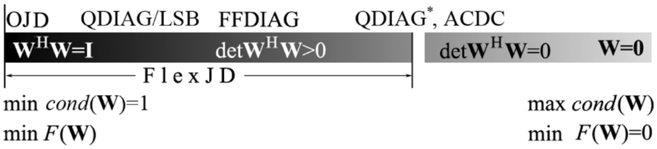


Fig. 4. Illustration of the relation of joint diagonalization algorithms when condition numbers of diagonalizers are concerned. QDIAG* denotes the QDIAG algorithm with $\mathbf{C}^{(0)} \notin \mathcal{R}$.

First, we would like to point out that, regarding the diagonalization error term $F(\mathbf{W})$, there are only global minima. See Proposition 3.

Lemma 1: $\mathrm{diag}\{\mathbf{X}^H[\mathbf{X} - \mathrm{Ddiag}(\mathbf{X})]\} = \mathbf{0}$ if and only if $\mathbf{X} - \mathrm{Ddiag}(\mathbf{X}) = \mathbf{0}$, i.e., $\mathbf{X}$ is a diagonal matrix.

Lemma 1 can be verified by straightforward calculations.

Proposition 3: $(\partial/\partial\mathbf{W})F(\mathbf{W}) = \mathbf{0}$ if and only if $F(\mathbf{W}) = 0$, where $F(\mathbf{W})$ is defined by (2).

The proof of Proposition 3 is presented in Appendix II.

From Proposition 3, $F(\mathbf{W})$ has only global minima. Its minima include trivial solutions, degenerate solutions, and if possible, nondegenerate solutions. To make the results reasonable, degenerate diagonalizers should be avoided strictly. Consequently, there are two situations in the joint diagonalization problem: 1) the target set $\mathcal{R}$ can be exactly jointly diagonalized by a nondegenerate matrix; and 2) the set cannot be exactly jointly diagonalized by any nondegenerate matrix. In the first situation, nondegenerate diagonalizers with zero diagonalization [i.e., $F(\mathbf{W}) = 0$] are achievable. In this situation, FlexJD is believed to be of global convergence [see Case 3 in $\mathrm{MOQO}(\widetilde{\mathbf{C}}_i, \mathbf{Q}_i)$]. This point is also illustrated by simulations. In the second situation, we can only reach a tradeoff between the zero diagonalization error and nondegeneracy of diagonalizers. Recall that model (7) is a multiobjective optimization model. Hence, some solutions achieve smaller diagonalization errors but have larger condition numbers; some others are just on the contrary. FlexJD attempts to obtain a kind of Pareto optimal solutions, which makes $G(\mathbf{W})/F(\mathbf{W})$ be maximized. This property will be illustrated by simulations.

For exactly jointly diagonalizable set of matrices, we further present the fact that the nondegenerate diagonalizers are essentially unique under some mild conditions.

Proposition 4: For given $\mathcal{R} = \{\mathbf{A}\mathbf{\Lambda}_k\mathbf{A}^H \in \mathbb{C}^{N\times N} | k \in \mathcal{I}_K\}$, where $K \geq 2$ and $\mathbf{\Lambda}_k$ are diagonal matrices, assume that there exist m, n such that $\mathbf{\Lambda} = \mathbf{\Lambda}_m^{-1}\mathbf{\Lambda}_n$ has distinct diagonal elements. Then, the diagonalizers of $\mathcal{R}$ are unique up to scale factors and a permutation of columns.

The proof of Proposition 4 is presented in Appendix III.

Proposition 4 is particularly significant for BSS. Taking SOS-based BSS methods into account, suppose that the sources have different spectrum. Hence, there exist two time lags τ_1 and τ_2 such that $\mathbf{R}_\mathbf{s}(\tau_1)^{-1}\mathbf{R}_\mathbf{s}(\tau_2)$ has distinct diagonal entries. Recall that $\mathbf{R}_\mathbf{x}(\tau) = \mathbf{A}\mathbf{R}_\mathbf{s}(\tau)\mathbf{A}^H$ ($\mathbf{R}_\mathbf{s}(\tau)$ are diagonal under the assumption of mutual independence of sources). If $\mathbf{W}$ is a diagonalizer of $\mathbf{R}_\mathbf{x}(\tau)$, from Proposition 4, $\mathbf{W}^H = \mathbf{A}^{-1}$ up to a permutation and scale factors. Hence, the sources are recovered by $\mathbf{y}(t) = \mathbf{W}^H\mathbf{x}(t) = \mathbf{W}^H\mathbf{A}\mathbf{s}(t)$ successfully.

IV. Simulations

In this section, we compare the performance of the FFDIAG algorithm, the ACDC algorithm, the QDIAG/LSB algorithm, and the Fajd algorithm via numerical experiments. There have been some performance indices to evaluate joint diagonalization algorithms, such as the *score index* and the *GLLC index* [5], [22]. However, these indices are not reliable if one or more matrices in $\mathcal{R}$ are degenerate (that is often the case in which the matrices in $\mathcal{R}$ are contaminated with noise). Since these algorithms are essentially based on criterion (2), we define the diagonalization error index (DEI) as follows:

$$\mathrm{DEI} = \frac{1}{KN(N-1)} \sum_{k\in\mathcal{I}_K} \mathrm{off}(\mathbf{W}^H\mathbf{R}_k\mathbf{W}) \qquad (10)$$

where K and N are used to eliminate the influence of scales of problems. The columns of diagonalizers obtained by each algorithm are normalized to be of unit norm. In each run, the matrices in $\mathcal{R}$ are generated as follows [5]:

$$\mathbf{R}_k = \mathbf{A}^H\mathbf{\Lambda}_k\mathbf{A} + \sigma^2\mathbf{n}_k^H\mathbf{n}_k \qquad (11)$$

where the matrix $\mathbf{\Lambda}_k$ is diagonal. The diagonal elements of $\mathbf{\Lambda}_k$ are drawn from the standard normal distribution, so are the elements of $\mathbf{A}$ and $\mathbf{n}_k$. The term $\sigma^2\mathbf{n}_k^H\mathbf{n}_k$ denotes the nondiagonalizable component (i.e., noise). The parameter σ allows one to control the level of noise.

1) *Joint diagonalization of exactly diagonalizable data sets.* Let $M = N = K = 5$ and $\sigma = 0$. Regardless of the initial value of $\mathbf{W}_0$ being $\mathbf{0}, \mathbf{I}$, or randomly generated matrices, FlexJD always achieves zero diagonalization error. Moreover, the diagonalizer $\mathbf{W}$ satisfies that $\mathbf{WA} = \mathbf{PD}$ (including that $\mathbf{A}$ is orthogonal), where $\mathbf{P}$ is a permutation matrix and $\mathbf{D}$ is a diagonal matrix. These results show that FlexJD is very likely to be of global convergence in this case.

2) *Joint diagonalization of approximately diagonalizable data sets.* Let $M = N = 10$, $\sigma = 10^{-6}, 10^{-5}, \ldots, 10^{-1}$, respectively. All the obtained results have been averaged over 100 Monte Carlo trials. They are shown in Fig. 3(a), which illustrates the evolution of the diagonalization errors (in logarithmic scales) versus the level of noise, and in Fig. 3(b), which illustrates the evolution of the condition numbers of diagonalizers (in logarithmic scales) versus the level of noise. From Fig. 3, FlexJD always achieves the smallest diagonalization error and, at the same time, maintains favorable condition numbers of diagonalizers. For $\sigma = 10^{-1}$, Fajd and FlexJD almost give the same result. We see again that the ACDC algorithm cannot avoid the degenerate solutions. (Because there is no guarantee that there exist positive-definite matrices in $\mathcal{R}$, QDIAG/LSB is excluded from this comparison.)

   From a purely mathematical point of view, OJD algorithms minimize the diagonalization error on condition that the smallest condition number is specified. QDIAG/LSB, as an approximate OJD algorithm, provides diagonalizers with larger condition numbers than those of OJD algorithms but smaller than those of others. FFDIAG relaxes the constraint of orthogonality significantly and generally yields well-conditioned diagonalizers. QDIAG with $\mathbf{C}^{(0)} \notin \mathcal{R}$ (denoted by QDIAG* for convenience) often achieves the smallest diagonalization error, but sometimes leads to degenerate solutions, as ACDC does. Otherwise, FlexJD prefers a favorable tradeoff: it neither goes unreasonably for a zero diagonalization error, nor does it excessively demand for a minimum condition number. Generally, if possible, FlexJD provides an exact nondegenerate diagonalizer. Otherwise, from the experimental results in Fig. 3, it gives a diagonalizer which minimizes the diagonalization error and the corresponding condition number simultaneously. Although both OJD algorithms and the FlexJD algorithm try to give Pareto optimal solutions to (7), FlexJD is more flexible. Relation of these algorithms is illustrated in Fig. 4.

   We also apply FFDIAG, FlexJD, and ACDC to five data sets with $\sigma = 10^{-1}$ to investigate their convergence rates. Fig. 5(a) shows the resulting diagonalization errors versus the number of iterations and Fig. 5(b) shows the corresponding condition numbers verses the number of iterations. From Fig. 5(a), we see that FlexJD converges slightly faster than FFDIAG and considerably faster than ACDC. The condition numbers of the diagonalizers given by FFDIAG and FlexJD are stable and small, but fluctuant and large for ACDC.

   FlexJD is able to provide nonsquare diagonalizers. As an example, let $M = 4$ and $N = 3$, $\sigma = 10^{-1}$. The matrices for diagonalization and the diagonalization results are shown in the first and second rows of Table I, respectively.

   The diagonalizer given by FlexJD is

$$\mathbf{W} = \begin{bmatrix} -0.2910 & 0.1827 & -0.2021 & 0.9171 \\ -0.2359 & -0.2044 & -0.6571 & -0.6861 \\ 0.4173 & -0.8332 & -0.3402 & 0.1257 \end{bmatrix}$$

   and $\mathrm{cond}(\mathbf{W}) \approx 1.73$. From Table I, FlexJD performs well again in nonsquare joint diagonalization.

3) *Applications in BSS.* We consider a slightly modified model of BSS [15]

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \delta\boldsymbol{\Delta}_0\mathbf{n}(t), \qquad t = 1, 2, \ldots, T \tag{12}$$

   where the elements of $\mathbf{A}$ are drawn from the standard normal distribution in each run, and $\boldsymbol{\Delta}_0 = \mathrm{Ddiag}(\mathbf{A}\mathbf{A}^H)$. δ is used to control the level of noise (nl) which is defined by $\mathrm{nl} = 10\log_{10}\delta^2$ (see [15] for detailed settings). Since the second-order blind identification (SOBI) algorithm [1] and the joint approximate diagonalization of eigenmatrices (JADE) algorithm almost cannot be improved in noiselessness cases ($\delta = 0$) [15], we mainly concentrate on the comparison when noise is involved. In each run, total of 20 correlation matrices are generated with the time lags $\tau_k = 1, 2, \ldots, 20$, respectively. All the obtained results have been averaged over 100 Monte Carlo trials.

The cross-talking error (CTE) is used to evaluate the estimate of the mixing matrix [17], [31]

$$\mathrm{CTE}(\mathbf{G}) = \frac{1}{2}\sum_i \left(\sum_j \frac{|\mathbf{G}_{ij}|}{\max\limits_l |\mathbf{G}_{il}|} - 1 \right) + \frac{1}{2}\sum_j \left(\sum_i \frac{|\mathbf{G}_{ij}|}{\max\limits_l |\mathbf{G}_{lj}|} - 1 \right) \tag{13}$$

where $\mathbf{G} = \mathbf{W}^H\mathbf{A}$ and $\mathbf{W}$ is the diagonalizer (i.e., the unmixing matrix). $\mathrm{CTE}(\mathbf{G}) = 0$ if and only if $\mathbf{G} = \mathbf{PD}$, where $\mathbf{P}$ is a permutation matrix and $\mathbf{D}$ is a diagonal matrix. Signal-to-noise ratio (SNR) is used to evaluate the estimate of each source signal

$$\mathrm{SNR}(s, y) = 10\log\frac{E[s^2]}{E\left[(y-s)^2\right]} \quad \mathrm{dB} \tag{14}$$

where s and y are random variables with 0-mean and 1-variance.

The first set of source signals consists of four speech signals. The sample length is 160 000. FlexJD is compared with Fajd, LSB, and SOBI this time. The observations are prewhitened to meet the requirement of the SOBI algorithm and the LSB algorithm (to other NJD algorithms, this procedure makes no difference except that they will treat a new mixing matrix $\widetilde{\mathbf{A}} = \mathbf{UA}$ instead of $\mathbf{A}$, where $\mathbf{U}$ is the whitening matrix). Only the LSB algorithm uses the zero-lag correlation matrix and it uses the output of SOBI as starting value. All results are averaged over 100 Monte Carlo trials. Fig. 6 shows the resulting CTE values (in logarithmic scales) at different level of noise. From Fig. 6, FlexJD achieves higher accuracy than that of the other algorithms.

Fig. 7 is an illustration of how condition numbers of the mixing matrix influence each algorithm. We define six levels of condition numbers as $[1, 10), [10, 10^2), \ldots, [10^5, 10^6)$, respectively. In each level, total 100 mixing matrices are generated

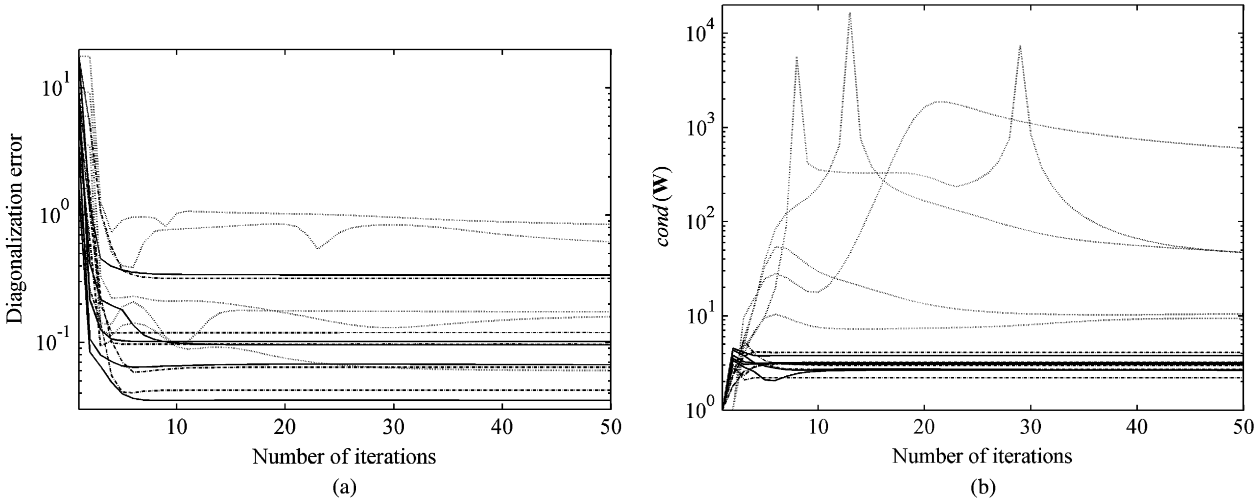


Fig. 5. Evolution of diagonalization errors (in logarithmic scales) and the condition numbers of diagonalizers versus the number of iterations for five different data sets, respectively. Solid lines: FlexJD. Dashed-dotted lines: FFDIAG. Dotted lines: ACDC.

TABLE I
EXAMPLE OF NONSQUARE JOINT DIAGONALIZATION. THE FIRST ROW SHOWS THE SET OF MATRICES FOR DIAGONALIZATION AND THE SECOND ROW SHOWS THE DIAGONALIZATION RESULTS

$\begin{bmatrix} 0.2482 & 0.1940 & -0.3762 & -0.4136 \\ 0.1940 & 0.9802 & -2.0721 & -0.4708 \\ -0.3762 & -2.0721 & 4.9370 & 0.8543 \\ -0.4136 & -0.4708 & 0.8543 & 0.7476 \end{bmatrix}$	$\begin{bmatrix} -0.9212 & -0.7990 & 1.1950 & 0.4014 \\ -0.7990 & -1.2378 & 2.8268 & 0.7260 \\ 1.1950 & 2.8268 & -6.1914 & -1.5361 \\ 0.4014 & 0.7260 & -1.5361 & -0.4936 \end{bmatrix}$	$\begin{bmatrix} -5.1541 & -3.0430 & 3.0641 & 0.2041 \\ -3.0430 & -2.4229 & 3.6469 & 0.5699 \\ 3.0641 & 3.6469 & -6.7886 & -1.2678 \\ 0.2041 & 0.5699 & -1.2678 & -0.5775 \end{bmatrix}$
$\begin{bmatrix} 0.7186 & -0.0000 & -0.0000 \\ -0.0000 & 2.3883 & 0.0000 \\ -0.0000 & 0.0000 & 0.0863 \end{bmatrix}$	$\begin{bmatrix} -0.1722 & 0.0000 & 0.0000 \\ 0.0000 & -3.0075 & -0.0000 \\ 0.0000 & -0.0000 & 0.0961 \end{bmatrix}$	$\begin{bmatrix} -0.3137 & -0.0000 & -0.0000 \\ -0.0000 & -2.8724 & 0.0000 \\ -0.0000 & 0.0000 & -0.0504 \end{bmatrix}$

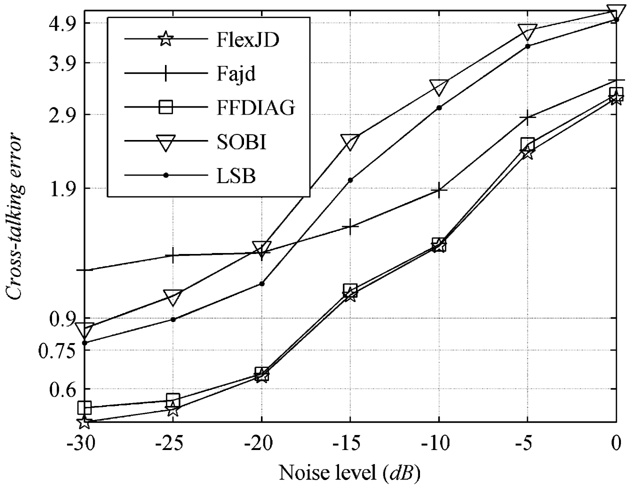


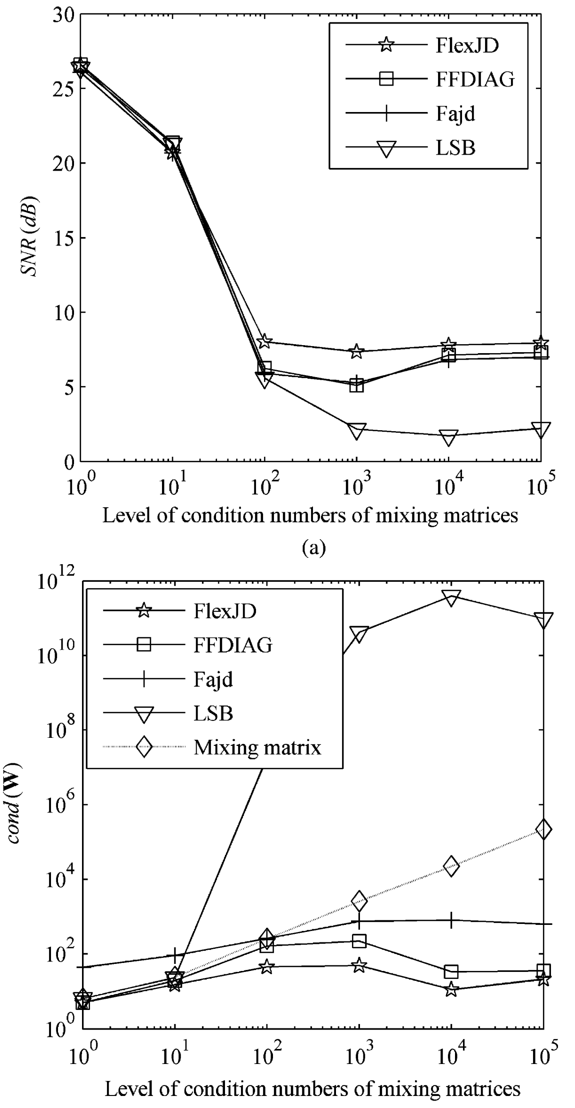
Fig. 6. Evolution of the cross-talking errors (in logarithmic scales) versus the level of noise in the separation of four speech signals.

(for example, in level one, every mixing matrix satisfies that $1 \leq \mathrm{cond}(\mathbf{A}) < 10$). A total of 50-dB white Gaussian noise is added to the observations. Mean SNR and mean condition numbers of diagonalizers versus the level of condition numbers of the mixing matrix are shown in Fig. 7(a) and (b), respectively. From Fig. 7(b), the condition numbers of the unmixing matrices estimated by FlexJD and FFDIAG are generally smaller than 10^2. Consequently, their performances in source separation are fairly stable when the condition number of the mixing matrix is greater than 10^2. When the condition number of the mixing matrix is greater than 10^2, FlexJD outperforms the others. The unmixing matrix estimated by LSB is a multiplication of the output diagonalizer and the whitening matrix. Because the whitening matrix is ill-conditioned, LSB performs badly when the mixing matrix has a large condition number. As a result, FlexJD is more robust to the noise involved in observations.

Finally, we consider the case in which $\mathcal{R}$ is complex. The sources are four 16-QAM digital communications signals and the number of samples is 1000. The real part and image part of the mixing matrix are drawn from independent normal distributions, respectively. In this experiment, cumulant matrices are used for diagonalization, and FlexJD, LSB, and JADE are compared (FFDIAG is absent because it currently cannot deal with complex-valued matrices). The output of JADE is used as starting value for LSB. In fact, FlexJD yields more reliable and preferable results again, as Fig. 8 shows.

V. CONCLUSION

In this paper, the joint diagonalization problem is reviewed as a multiobjective optimization problem for the first time. Based on this, a new NJD algorithm, named FlexJD, is developed. FlexJD yields flexible diagonalizers which not only minimize



(a)

(b)

Fig. 7. (a) Evolution of the SNR versus the level of condition numbers of the mixing matrices. (b) Evolution of the condition numbers of the unmixing matrix (in logarithmic scales) versus the level of condition numbers of the mixing matrices.

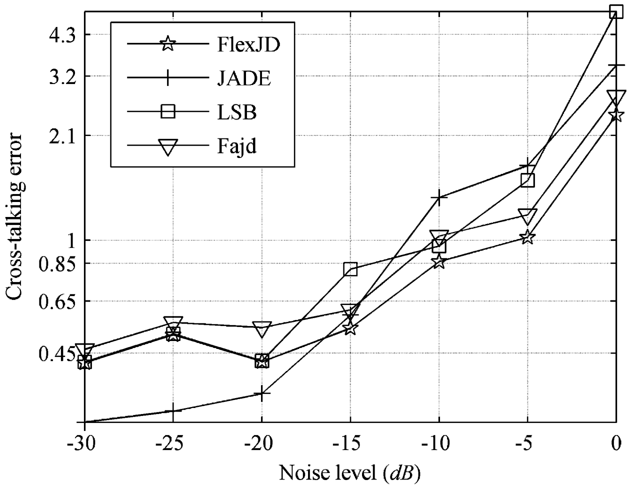


Fig. 8. Evolution of the cross-talking errors (in logarithmic scales) versus the level of noise in the separation of four 16-QAM digital communications signals.

the diagonalization error, but also have as small condition numbers as possible. Meanwhile, degenerate solutions are avoided strictly. FlexJD is also one of the most versatile joint diagonalization algorithms: it does not require the set of matrices to be positive definite, real, or symmetric, and the diagonalizers even can be nonsquare.

The block coordinate descent method is used to optimize the model. This way is proved to be simple and efficient. The convergence is also discussed under the condition of $M = N$. If the set of matrices is exactly jointly diagonalizable, we prove that there are only global minima for the joint diagonalization problem and the solutions are unique under mild conditions. The uniqueness of solutions is also significant to BSS.

Generally, FFDIAG performs well but only applicable to real matrices sets so far. QDIAG (including ACDC) often yields degenerate solutions except that the first matrix is positive definite, as the LSB algorithm states. Compared with these algorithms, the FlexJD algorithm is more efficient, widely applicable, and easy to implement.

Appendix I
Proof of Proposition 2

Proof:

1) Let $\mathbf{W} = [\mathbf{w}_i\ \overline{\mathbf{W}_i}]\mathbf{P}_{1i}$, where $\mathbf{P}_{1i}$ is a permutation matrix. Thus, we have

$$\begin{aligned}
&\left|\det(\mathbf{W}^H\mathbf{W})\right| \\
&= \left|\det\left(\mathbf{P}_{1i}^H \begin{bmatrix} \mathbf{w}_i^H \\ \overline{\mathbf{W}_i}^H \end{bmatrix} [\mathbf{w}_i \quad \overline{\mathbf{W}_i}]\mathbf{P}_{1i}\right)\right| \\
&= \left|\det\begin{bmatrix} \mathbf{w}_i^H\mathbf{w}_i & \mathbf{w}_i^H\overline{\mathbf{W}_i} \\ \overline{\mathbf{W}_i}^H\mathbf{w}_i & \overline{\mathbf{W}_i}^H\overline{\mathbf{W}_i} \end{bmatrix}\right| \\
&= \det\left(\overline{\mathbf{W}_i}^H\overline{\mathbf{W}_i}\right) \\
&\quad \times \det\left(\mathbf{w}_i^H\mathbf{w}_i - \mathbf{w}_i^H\overline{\mathbf{W}_i}\left(\overline{\mathbf{W}_i}^H\overline{\mathbf{W}_i}\right)^{-1}\overline{\mathbf{W}_i}^H\mathbf{w}_i\right) \\
&= \det\left(\overline{\mathbf{W}_i}^H\overline{\mathbf{W}_i}\right)\mathbf{w}_i^H\left(\mathbf{I} - \overline{\mathbf{W}_i}\left(\overline{\mathbf{W}_i}^H\overline{\mathbf{W}_i}\right)^{-1}\overline{\mathbf{W}_i}^H\right)\mathbf{w}_i.
\end{aligned}$$

2) Since $\dim(\mathrm{col}(\overline{\mathbf{W}_i})) + \dim(\overline{\mathbf{W}_i}^{\perp}) = N$ and $\mathbf{C}_i$ is an orthogonal basis of $\overline{\mathbf{W}_i}^{\perp}$, we suppose that $\mathbf{w}_i = \mathbf{C}_i\mathbf{x} + \overline{\mathbf{W}_i}\mathbf{y}$. Substitute $\mathbf{w}_i$ to $\mathbf{w}_i^H\mathbf{w}_i = 1$, and together with $\mathbf{C}_i^H\overline{\mathbf{W}_i} = \mathbf{0}$, we have $1 = \mathbf{x}^H\mathbf{x} + \mathbf{y}^H\overline{\mathbf{W}_i}^H\overline{\mathbf{W}_i}\mathbf{y}$. Note that $\mathbf{w}_i^H(\mathbf{I} - \overline{\mathbf{W}_i}(\overline{\mathbf{W}_i}^H\overline{\mathbf{W}_i})^{-1}\overline{\mathbf{W}_i}^H)\mathbf{w}_i = 1 - \mathbf{y}^H\overline{\mathbf{W}_i}^H\overline{\mathbf{W}_i}\mathbf{y}$, $\mathbf{w}_i^H\mathbf{C}_i\mathbf{C}_i^H\mathbf{w}_i = \mathbf{x}^H\mathbf{x}$. The result yields immediately.

Appendix II
Proof of Proposition 3

Proof: Write $\mathbf{X}_k = \mathbf{W}^H\mathbf{R}_k\mathbf{W}$, $k \in \mathcal{I}_K$. Note that $(\partial/\partial\mathbf{W})F(\mathbf{W}) = \sum_{k\in\mathcal{I}_K} 4\mathbf{R}_k\mathbf{W}[\mathbf{X}_k - \mathrm{Ddiag}(\mathbf{X}_k)]$. Therefore, if $F(\mathbf{W}) = 0$, $(\partial/\partial\mathbf{W})F(\mathbf{W}) = \mathbf{0}$ yields immediately.

Conversely, assume that $(\partial/\partial\mathbf{W})F(\mathbf{W}) = 0$. Then, $\mathbf{W}^H(\partial/\partial\mathbf{W})F(\mathbf{W}) = \mathbf{0}$. From $\mathrm{diag}\{\mathbf{X}_k^H[\mathbf{X}_k - \mathrm{Ddiag}(\mathbf{X}_k)]\} \geq \mathbf{0}$, we have $\mathrm{diag}\{\mathbf{X}_k^H[\mathbf{X}_k - \mathrm{Ddiag}(\mathbf{X}_k)]\} = \mathbf{0}$. According to Lemma 1, $\mathbf{X}_k - \mathrm{Ddiag}(\mathbf{X}_k) = \mathbf{0}$, i.e., $F(\mathbf{W}) = 0$. □

Appendix III
Proof of Proposition 4

Proof: Suppose $\mathbf{W}$ is a diagonalizer of $\mathcal{R}$. Write $\mathbf{W}^H\mathbf{R}_k\mathbf{W} = \mathbf{\Gamma}_k$ and $\mathbf{\Gamma} = \mathbf{\Gamma}_m^{-1}\mathbf{\Gamma}_n$, where $\mathbf{\Gamma}_k$ are diagonal matrices, $k \in \mathcal{I}_K$. Let $\mathbf{C} = \mathbf{A}^H\mathbf{W}$. We have $\mathbf{\Gamma}_k = \mathbf{C}^H\mathbf{\Lambda}_k\mathbf{C}$ for $k \in \mathcal{I}_K$. Thus, from $\mathbf{\Gamma} = \mathbf{\Gamma}_m^{-1}\mathbf{\Gamma}_n$, $\mathbf{\Gamma} = \mathbf{C}^{-1}\mathbf{\Lambda}_m^{-1}\mathbf{\Lambda}_n\mathbf{C}$ yields immediately, i.e.,

$$\mathbf{C\Gamma} = \mathbf{\Lambda C} \tag{15}$$

or, equivalently

$$c_{ij}\gamma_j = \lambda_i c_{ij}, \quad \forall i, j \in \mathcal{I}_M. \tag{16}$$

Since $\mathbf{C}$ is invertible, there exists at least one nonzero element in each column of $\mathbf{C}$. For any column k, without loss of generality, suppose that $c_{ik} \neq 0$. From (16), $\gamma_k = \lambda_i$ yields. Assume to arrive at a contradiction that there is another nonzero element in the kth column of $\mathbf{C}$, namely, $c_{lk} \neq 0$. From (16) again, $\gamma_k = \lambda_l$. Thus, $\lambda_i = \lambda_l$, which is a contradiction. Consequently, there is one and only one nonzero element in each column of $\mathbf{C}$. From the invertibility of $\mathbf{C}$, $\mathbf{C}^H = \mathbf{W}^H\mathbf{A} = \mathbf{PD}$ holds, where $\mathbf{P}$ is a permutation matrix and $\mathbf{D}$ is a diagonal matrix. □

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that have led to the present improved version of the original manuscript.

References

[1] A. Belouchrani, K. AbedMeraim, J. F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, Feb. 1997.

[2] A. Yeredor, "Blind source separation via the second characteristic function," *Signal Process.*, vol. 80, no. 5, pp. 897–902, 2000.

[3] E. Moreau, "A generalization of joint-diagonalization criteria for source separation," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 530–541, Mar. 2001.

[4] M. John and K. Rahbar, "Joint diagonalization of correlation matrices by using newton methods with application to blind signal separation," in *Proc. IEEE Sensor Array Multichannel Signal Process. Workshop*, 2002, pp. 403–407.

[5] A. Ziehe, P. Laskov, G. Nolte, and K. R. Muller, "A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation," *J. Mach. Learn. Res.*, vol. 5, pp. 777–800, 2004.

[6] F. X. Wang, Z. K. Liu, and J. Zhang, "Nonorthogonal joint diagonalization algorithm based on trigonometric parameterization," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5299–5308, Nov. 2007.

[7] G. Chabriel, J. Barrere, N. Thirion-Moreau, and E. Moreau, "Algebraic joint zero-diagonalization and blind sources separation," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 980–989, Mar. 2008.

[8] F. X. Wang, Z. K. Liu, and J. Zhang, "A new joint diagonalization algorithm with application in blind source separation," *IEEE Signal Process. Lett.*, vol. 13, no. 1, pp. 41–44, Jan. 2006.

[9] L. De Lathauwer and J. Castaing, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1096–1105, Mar. 2008.

[10] B. N. Chen and A. P. Petropulu, "Frequency domain blind MIMO system identification based on second- and higher order statistics," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1677–1688, Aug. 2001.

[11] J. F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *Proc. Inst. Electr. Eng.—Radar Signal Process. F*, vol. 140, no. 6, pp. 362–370, 1993.

[12] J. Sheinvald, "On blind beamforming for multiple non-Gaussian signals and the constant-modulus algorithm," *IEEE Trans. Signal Process.*, vol. 46, no. 7, pp. 1878–1885, Jul. 1998.

[13] E. Moreau and P. Comon, "Comments on blind beamforming for multiple non-Gaussian signals and the constant-modulus algorithm," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3248–3250, Nov. 2000.

[14] E. Fadaili, N. T. Moreau, and E. Moreau, "Nonorthogonal joint diagonalization/zero diagonalization for source separation based on time-frequency distributions," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1673–1687, May 2007.

[15] S. Degerine and E. Kane, "A comparative study of approximate joint diagonalization algorithms for blind source separation in presence of additive noise," *IEEE Trans. Signal Process.* , vol. 55, no. 6, pt. 2, pp. 3022–3031, Jun. 2007.

[16] A. Hyvarinen, J. Karhunen, and E. Oja*, Independent Component Analysis*. New York: Wiley, 2001.

[17] A. Cichocki and S. Amari*, Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. New York: Wiley, 2002.

[18] J. F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 1, pp. 161–164, 1996.

[19] D. T. Pham, "Joint approximate diagonalization of positive definite Hermitian matrices," *SIAM J. Matrix Anal. Appl.*, vol. 22, no. 4, pp. 1136–1152, 2001.

[20] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1545–1553, Jul. 2002.

[21] R. Vollgraf and K. Obermayer, "Quadratic optimization for simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3270–3278, Sep. 2006.

[22] X. L. Li and X. D. Zhang, "Nonorthogonal joint diagonalization free of degenerate solution," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1803–1814, May 2007.

[23] J. F. Cardoso, "On the performance of orthogonal source separation algorithms," presented at the Europ. Assoc. Signal Process., Edinburgh, U.K., 1994.

[24] S. Dégerine, "Sur la diagonalisation conjointe approchée par un critère des moindres carrés," presented at the Groupe de Recherche et d'Etudes de Traitement du Signal et des Images (GRETSI) Conf., Toulouse, France, 2001.

[25] H. W. Guggenheimer, A. S. Edelman, and C. R. Johnson, "A simple estimate of the condition number of a linear system," *College Math*, vol. 26, no. 1, pp. 2–5, 1995.

[26] J. K. Merikoski, U. Urpala, A. Virtanen, T. Y. Tam, and F. Uhlig, "A best upper bound for the 2-norm condition number of a matrix," *Linear Algebra Appl.*, vol. 254, pp. 355–365, 1997.

[27] W. Stadler, "A survey of multicriteria optimization or the vector maximization problem: Part I: 1776–1960," *J. Optim. Theory Appl.*, vol. 29, no. 1, pp. 1–52, 1979.

[28] W. Stadler, "A comprehensive bibliography on multicriteria decision making and related areas," Univ. California, Berkeley, CA, 1981.

[29] A. N. Tiknonov and V. Y. Arsenin*, Solutions of Ill-Posed Problems*. Washington, DC: Wiley, 1977.

[30] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Norwell, MA: Kluwer, 2003.

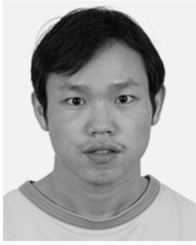
[31] M. Girolami*, Self-Organizing Neural Networks: Independent Component Analysis and Blind Source Separation*. London, U.K.: Springer-Verlag, 1999.

Guoxu Zhou was born in Hubei Province, China, in 1977. He is currently working towards the Ph.D. degree at the Intelligent Information and Signal Processing Group (led by Prof. S. Xie), South China University of Technology, Guangzhou, China.

His research interests include machine learning, intelligent information processing, and blind signal processing.

Shengli Xie (M'01–SM'02) was born in Hubei Province, China, in 1958. He received the M.S. degree in mathematics from Central China Normal University, Wuhan, China, in 1992 and the Ph.D. degree in control theory and applications from South China University of Technology, Guangzhou, China, in 1997.

Currently, he is a Full Professor with the South China University of Technology and a Vice Head of the Institute of Automation and Radio Engineering. He is the author or coauthor of two books and more than 70 scientific papers in journals and conference proceedings. His research interests include automatic control and blind signal processing.



Zuyuan Yang was born in Hubei Province, China, in 1982. Currently, he is working towards the Ph.D. degree at the Intelligent Information and Signal Processing Group (led by Prof. S. Xie), South China University of Technology, Guangzhou, China.

His research interests include information security, blind signal processing, and machine learning.

Jun Zhang received the B.S., M.Sc., and Ph.D. degrees from Beihang University, Beijing, China, in 1987, 1991, and 2001, respectively.

Currently, he is a Professor and Dean in the School of Electronics and Information Engineering, Beihang University. His current research interests are in the areas of integrated networks of air, ground, and sky communication systems, modern air traffic management, *ad hoc* networks, and network management. He authored two books, more than 60 referred papers, as well as more than ten patents or pending applications.

Dr. Zhang currently serves on the editorial boards of the *CAAI Transactions on Intelligent Systems* and two committees of international conferences.